Discrete Event Simulation of the DiffServ-over-MPLS with NIST GMPLS Lightwave Agile Switching Simulator (GLASS)

Youngtak Kim°*, Eunhyuk Lim*, Douglas Montgomery**

*Dept. of Information and Communication Engineering, Graduate School, YeungNam Univ.
214-1, Dae-Dong, KyungSan-Si, KyungBook, 712-749, KOREA
(Tel: 053-810-2497; Fax: 053-814-5713; E-mail: ytkim@yu.ac.kr)

**Advanced Network Technologies Division (ANTD),
National Institute of Standards and Technology (NIST)
820 West Diamond Avenue, Gaithersburg, MD 20899, U.S.A.

Abstract **–** In this paper, we explain the discrete event network simulation for DiffServ-aware-MPLS on the GMPLS-based WDM Optical Network. The simulation has been performed on the *NIST GMPLS Lightpath Agile Switching Simulator (GLASS).* We also briefly explain the design and implementation of the GLASS, and report the current implementation status and experimental results. The NIST GLASS has been developed to support the R&D works in the area of Next Generation Internet (NGI) networking with GMPLS-based WDM optical network and Internet traffic engineering with DiffServ-over-MPLS. It supports various discrete-event simulations of DiffServ packet classification, per-hop-behavior (PHB) processing with class-based-queuing, MPLS traffic engineering, MPLS OAM functions that provide performance monitoring and fault notification, GMPLS-based signaling for WDM optical network, fiber/lambda optical switching, link/node failure model, and fast fault detection, notification and restoration from an optical link failure. In this paper, we focus the discussions on the discrete event simulation of DiffServ-over-MPLS in NIST GLASS.

Keywords – MPLS/GMPLS, WDM Optical Network, DiffServ, Network Simulation, Traffic Engineering

I. Introduction

A. Motivation

In order to manage the explosively increasing Internet traffic more effectively, various traffic engineering and networking technologies have been proposed, developed and implemented. The physical link bandwidth has been expanded with DWDM optical transmission technology and Optical Add-Drop (OADM) & Optical Cross Connect (OXC) switching technologies [1]. MPLS (Multi-Protocol Label Switching) has been introduced to enhance the packet forwarding & switching performance by using faster fixed-label switching at layer 2.5 [2]. By using the connection-oriented, bandwidth reserved MPLS LSP (Label Switched Path) among the core routers, the traffic engineering has been more flexible and predictable. MPLS architecture, which had been basically designed upon packet switching capability, recently has been generalized into Generalized MPLS (GMPLS) to include other switching capabilities, such as TDM circuit switching, fiber/lambda switching with generalized label [3]. The implementation of IP-based control plane for the next generation optical network with the GMPLS control architecture has been received great interests recently; and it has been accepted by the optical network equipment vendors and network operators. The DiffServ technology has been developed to provide differentiated quality-of-service (QoS) according to the user's requirements[4]. Especially, the protocol structure of *DiffServ-over-MPLS on the GMPLS-based WDM Optical Network* has been emphasized as a promising technical solution for Next Generation Internet [5].

In order to test and evaluate the inter-operability and effectiveness of the newly proposed protocol functions, the network simulations with the configurable node protocol structure and the scalable network size have been used in popular by many researchers and system developer as the more practical approach. Network Simulator (ns) [6], JavaSim [7], SSFNet [8], and OPNET [9] are the most popularly used network simulators. But, these network simulators do not support the integrated simulation of "DiffServ-over-MPLS" on the "GMPLS-based WDM Optical Network" with OAM functions and fault restoration functions.

B. Network Simulation for DiffServ-over-MPLS on the GMPLS-based WDM Optical Network

The NIST GLASS has been developed for the integrated simulations of Next Generation Internet (NGI) networking with the GMPLS-based WDM optical network, and the Internet traffic engineering with DiffServ-over-MPLS [10]. It supports the discrete-event simulations of various DiffServ packet classification, per-hop-behavior (PHB) processing with class-based-queuing, MPLS traffic engineering, MPLS OAM for performance monitoring and fault restoration, GMPLS-based signaling for WDM optical network, link/node failure model, fiber/lambda optical switching, and fast fault detection, notification and restoration from link or node failure.

NIST GLASS is implemented on the SSFNet (Scalable Simulation Framework Network) simulation platform. It has been designed and implemented with open interfaces to support future expansion or replacement of protocol modules by users. It uses DML (domain modeling language) description input file interface to support the user's flexible definition/modification of simulation parameters and configuration of protocol modules.

The rest of this paper is organized as follows. Section II explains the architecture of NIST GMPLS Simulator for the discrete event simulation of DiffServ-over-MPLS, and Section III shows some simple network simulations, and analyzes the results. Section IV summarizes the paper.

II. Architecture of NIST GMPLS Networking Simulation Tool

A. DiffServ packet processing

For the differentiated or classified processing of packets at each IP routers, DiffServ architecture has been proposed and IETF documents define the differentiated service code points (DSCP), DiffServ class-types, metering and coloring, class-based-queuing with algorithmic packet drop, packet scheduling, and optional traffic shaping. Figure 1 shows the overall DiffServ per-class packet processing.

The packet classification is implemented with multi-field classifier that uses multiple fields in the IP packet header to determine the differentiated per-hop-behavior (PHB). The IP source address prefix (address and prefix length), destination address prefix, upper-layer protocol, TCP/UDP/SCTP source and destination port range, and ToS (Type of Service) or DSCP (DiffServ Code Point) fields are used in the packet classifier.

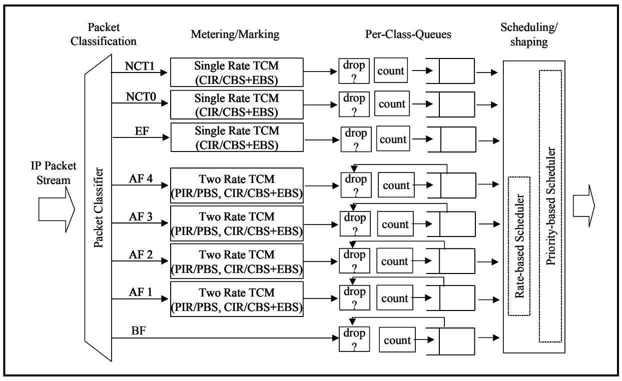


Figure 1. DiffServ Per-class Packet Processing

The DiffServ class-types are defined according to the performance objectives of end-user service traffic. The DiffServ class-types that are proposed in IETF can be grouped into 4 categories : network control traffic (NCT), expedited forwarding (EF), assured forwarding (AF), and best-effort (BE) forwarding. In the NIST GLASS, 8 class-types, are defined and their class-based-queuing mechanisms are specified in the DML file.

In order to protect the premium or higher priority traffic flow from the network congestion, the packet flow must be firstly measured according to the traffic parameters allocated to each class-type. For the measurements of the arrival intervals of packet, Token Bucket Meter (TBM) or Time Sliding Window (TSW) meters are used with single rate three color marker (SRTCM) or two rates three color marker (TRTCM). In the simulator, TBM with SRTCM is used for NCT and EF class-type where the packet rate is defined by peak information rate (PIR) with peak burst size (PBS), while TBM with TRTCM is used for AF class-type where the packet rate is defined by PDR/PBS and committed information rate (CIR) with committed burst size (CBS). According to the result of the data rate measurement, each packet is colored to Green (conforming PIR/PBS and CIR/CBS), Yellow (conforming PIR/PBS and CIR, but exceeding CBS), or Red (exceeding PIR/PBS).

The class-based-queuing functions include packet discarding according to the drop precedence and priority of the class-type, and packet buffering. Packet dropping at each class-base-queue is implemented with either simple tail-dropping or with more complex algorithmic random dropping as in RED (Random Early Detection) or RIO (RED with In/Out-Profile). These three dropping mechanisms are provided in the NIST GLASS. For the algorithmic random dropping, the smoothed queue lengths of each class-base-queue are continuously measured with the exponentially weighted moving average calculation.

The packet scheduler determines the selection of a packet to be transmitted. The packet is selected according to the priority of the queue (in priority-scheduler) or according to the relative weight of the queue (in weighted scheduler). In priority scheduling, the queue(s) with higher-priority exclusively use the bandwidth regardless of the lower-priority queue status. In weighted scheduler, weight for each queue is allocated, and the relative portion of the bandwidth is allocated the to the queue by the weighted round robin (WRR) or weighted fair queuing (WFQ). Also, various combinations of the priority scheduler and the weighted scheduler are possible. For example, we can use the WFQ for the all AF traffic flows with specific weight for each AF class-type, while the overall scheduling is handled by priority scheduler, as shown in Figure 2.

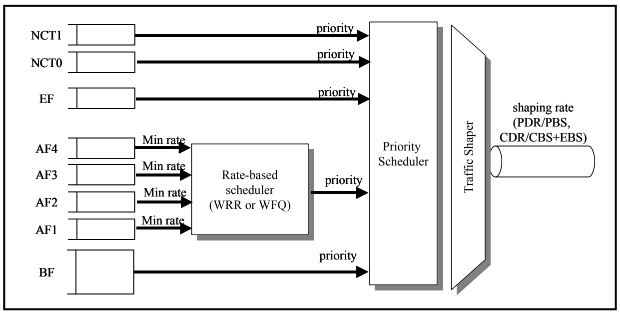


Figure 2. DiffServ Packet Scheduling

C. MPLS LSR

MPLS networking is based on the explicit connection setup and bandwidth management with signaling protocol (CR-LDP or RSVP), routing protocol (OSPF or IS-IS, BGP), and OAM (Operation, Administration and Maintenance).

Figure 3 shows the protocol organization of MPLS-LSR in the NIST GMPLS Simulator.

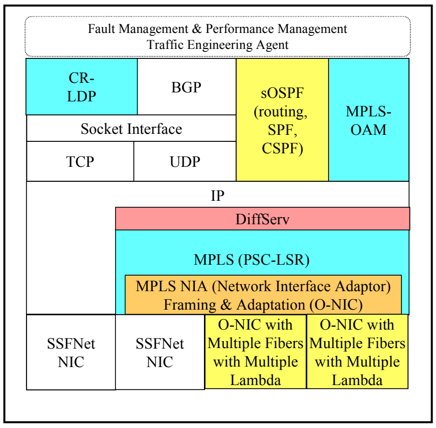


Figure 3. MPLS-LSR in the NIST GMPLS Simulator

The primary traffic engineering capability offered by MPLS is the ability to configure constraint-route label switched path (CR-LSP) routes with the traffic engineering constraints associated with the LSP. The ability to set up explicit routes with QoS constraints can be supported by one of two signaling protocols: RSVP-TE (resource ReSerVation Protocol with traffic engineering extension) or constraint-route label distribution protocol (CR-LDP). In the NIST GLASS, the CR-LDP MPLS signaling has been implemented first, and recently RSVP-TE also has been implemented. The traffic parameters TLV (type-length-value) of CR-LDP are Peak Data Rate (PDR), Peak Burst Size (PBS), Committed Data Rate (CDR), Committed Burst Size (CBS), and Excess Burst Size (EBS). Additional TE constraints, such as backup path type (1:1, 1+1, 1:N, M:N, link-disjoint or path-disjoint, SRLG (shared risk link group)-disjoint), resource color, and residual error, are defined as additional TLV in the CR-LDP signaling message and processed by the LSRs.

Traffic policing of CR-LDP is necessary to guarantee the traffic engineering constrains associated with the CR-LSP by preventing any unauthorized overuse of link resources. To measure the bandwidth utilization by each CR-LSP, dual token bucket meter is used; a token bucket for checking peak data rate (PDR) and PBS, and another token bucket for checking committed data rate (CDR) with CBS and EBS. According to the measurement result and the bandwidth over-utilization policy, the excess packets may be discarded, or the excess packets may be tagged and the dropping-decision is transferred to the upper-level traffic policing function of the outer tunnel LSP that may allow temporal over-utilization if there is un-used available bandwidth by the under-utilization of other CR-LSPs.

MPLS packet scheduling at the output ports can be implemented with similar structure of DiffServ packet scheduler that was explained in previous section.

III. Simulation Results and Analysis

A. Network Configuration of DiffServ-over-MPLS Simulation

Diffserv-over-MPLS provides the capability of the micro-flow traffic engineering for each class-type in a aggregated packet stream with a LSP. Figure 4 shows a simple network topology to test the functions of DiffServ-over-MPLS traffic engineering on network simulator. The DiffServ packet flow between host pairs 100-101, 102-103, and 104-105 are supported by a LSP between LER 110-LER111. Also a LSP between LER150-LER151 supports the three DiffServ packet flows of host 150-151, 152-153, and 154-155.

In order to test the guaranteed provisioning of bandwidth and QoS parameters, the simulation topology has bottleneck link between LSR 220 and LSR 221 through which all LSPs between LER pairs pass. The traffic generations also have different timing to simulate a gradually fluctuating network condition. The link capacities are configured to be 110% of the sum of the CDR of the DiffServ packet generation rate; in other word, the link utilization is 95%.

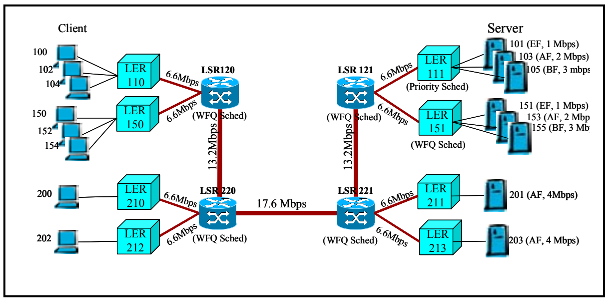
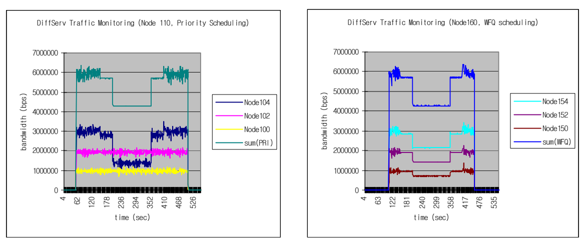


Figure 4. Simulation topology for DiffServ-over-MPLS

B. Guaranteed QoS provisioning

1) Bandwidth provisioning with priority-based scheduling and weight-based scheduling

Figure 5 compares the monitored bandwidths of each DiffServ class-type with different DiffServ packet scheduling mechanism. In both cases, the total bandwidth is maintained to be 6 Mbps on average when there is no congestion on the bottleneck link.



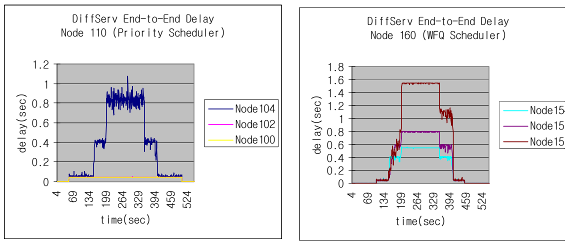
(a) Priority Scheduling (b) WFQ scheduling

Figure 5. Bandwidth monitoring of DiffServ traffic

When node 201 and 203 start to generate packet flow at 150~400 second and 200~350 second, traffic congestion is occurred on the bottleneck link, and the total traffic is reduced to the CDR of LSP (4.4 Mbps). In priority scheduling, as shown in Figure 5-(a), the used bandwidth of the class-type packet flows with higher priorities are maintained even in the congestion status. The lowest priority traffic (BF) is only affected, and the data rate is reduced under congestion. In weight-based scheduling, however, the three bandwidths are all readjusted according to the available total bandwidth. The relative sharing of the available bandwidth is controlled by the weight of each flow.

2) Guaranteed end-to-end delay and jitter

Figure 6 shows the measured end-to-end delay of each DiffServ packet flow. In priority scheduling the higher-priority packet flows are not affected by the congestion status. In weight-based scheduling, each packet flows are experiencing gradually increasing end-to-end delay as the traffic increases through the bottleneck link.

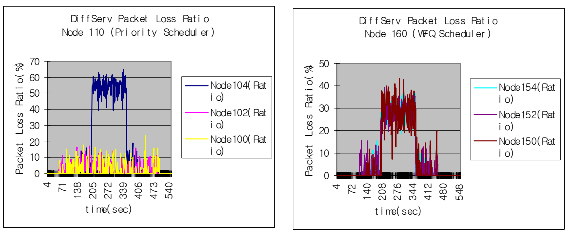


(a) Priority Scheduling (b) WFQ scheduling

Figure 6. End-to-End delay

3) Packet Loss Ratio

Figure 7 shows the packet loss ratio at each DiffServ packet scheduling. As we expected, the priority scheduling protects higher-priority packet flows, while the weight-based scheduling provides the relative share of available bandwidth. The measured packet loss ratio exactly reflect the basic operational properties of the priority-based scheduling and weight-based scheduling.



(a) Priority Scheduling (b) WFQ scheduling

Figure 7. Packet loss ratio

C. Optimal Bandwidth Utilization among LSPs

From Figure 5~7, we showed that the GLASS supports the mechanism of the bandwidth borrowing among LSPs when there is excess available bandwidth. The MPLS LSR/LER supports the re-distribution of excess bandwidth among the active working LSPs and the packet scheduling of DiffServ also receives this re-adjusted excess bandwidth information. By this mechanism, we can provide the optimal bandwidth utilization across the network.

IV. Conclusion

In this paper, we explained the discrete event simulation of DiffServ-over-MPLS with a GMPLS-based Optical Internet simulator, called *GLASS (GMPLS Lightwave Agile Switching Simulator)*. The GLASS has been developed to support the R&D works in the area of Next Generation Internet (NGI) networking with GMPLS-based WDM optical network, and Internet traffic engineering with DiffServ-over-MPLS.

In this paper, we focused the functions of NIST GLASS for DiffServ-over-MPLS, and analyzed the experimental simulation results of the DiffServ-over-MPLS packet processing. The bandwidth utilization, end-to-end delay, jitter and packet loss ratio have been measured and analyzed under varying traffic condition.

We didn't analyzed, in this paper, the measured data with comparisons of M/D/1 and ND/D/1 queuing mathematical model for the DiffServ packet queuing & scheduling and the MPLS packet queuing & scheduling, respectively. This mathematical analysis is under study, and will be reported by other paper.

References

[1] Antonio Rodrigues-Moral et al., "Optical Data Networking: Protocols, Technologies, and Architectures for Next Generation Optical Transport Networks and Optical Internetworks," Journal of Lightwave Technology, Vol. 18, No. 12, December 2000, pp. 1855~1870.
[2] Jeremy Lawrence, "Designing Multiprotocol Label Switching Networks," IEEE Communications Magazine, July 2001, pp. 134~142.
[3] Ayan Banerjee et al., "Generalized Multiprotocol Label Switching (GMPLS): An Overview of Routing and Management Enhancements," IEEE Communications Magazine, July 2001, pp. 144~151.
[4] Panos Trimintzios et al., "A Management and Control Architecture for Providing IP Differentiated Services in MPLS-Based Networks," IEEE Communications Magazine, May 2001, pp. 80~88.
[5] Xipeng Xiao et al., "Traffic Engineering with MPLS in the Internet," IEEE Network, March/April 2000, pp. 28~33.
[6] The Network Simulator (ns), http://www.isi.edu/nsnam/ns/
[7] JavaSim, http://eepc117.eng.ohio-state.edu/javasim
[8] SSF (Scalable Simulation Foundation), http://www.ssfnet.org/exchangePage.html
[9] OPNET Modeler, http://www.opnet.com/
[10] NIST GMPLS/Optical Internet Simulator, http://www.antd.nist.gov/glass/